(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 November 2000 (30.11.2000)

PCT

(10) International Publication Number
WO 00/72583 A1

(51) International Patent Classification[7]: H04N 5/00

(21) International Application Number: PCT/US00/07980

(22) International Filing Date: 24 March 2000 (24.03.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/135,221 21 May 1999 (21.05.1999) US

(71) Applicant *(for all designated States except US)*: GENERAL INSTRUMENT CORPORATION [US/US]; 101 Tournament Drive, Horsham, PA 19044 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: BIRNBAUM, Jack, M. [US/US]; 559 Nicole Drive, Southampton, PA 18966 (US). DEL SORDO, Chris [US/US]; 229 Heatherfield Drive, Souderton, PA 18964 (US). GAZDA, Robert [US/US]; 42 Le Forge Court, Wayne, PA 19087 (US).

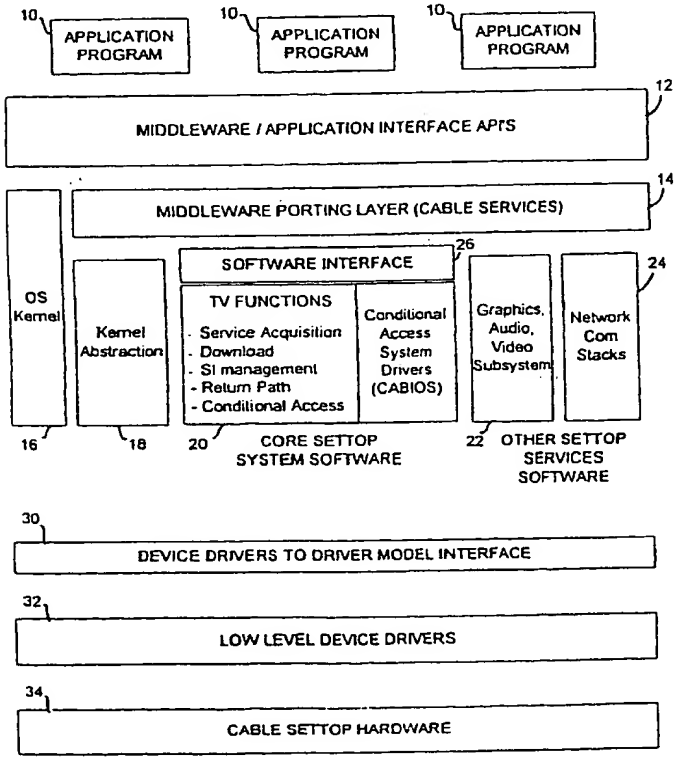
(74) Agent: LIPSITZ, Barry, R.; 755 Main Street, Building No. 8, Monroe, CT 06468 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: SOFTWARE ARCHITECTURE FOR A TELEVISION SET-TOP TERMINAL PROVIDING COMPATIBILITY WITH MULTIPLE OPERATING ENVIRONMENTS



(57) Abstract: A software architecture is provided to enable core television set-top software (20) to run in different operating environments. A kernel abstraction component (18) uses kernel calls and kernel-specific translations to abstract operating system-specific functions from an operating system kernel (16). The functions are made available to the core software (20) in a generic format. Additionally, a device drivers-to-driver model interface (30) separates the details of the driver model environment from the core system software (20). Furthermore, a software interface (26) allows application programs (10) (such as program guides, games, and Internet web browsers) and middleware (12) to communicate with the core set-top system software (20). The core software allows the set-top to provide television functions such as service acquisition, system information management, download capability, return path communication, set-top configuration, and conditional access control.

WO 00/72583 A1

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SOFTWARE ARCHITECTURE FOR A TELEVISION SET-TOP TERMINAL PROVIDING COMPATIBILITY WITH MULTIPLE OPERATING ENVIRONMENTS

BACKGROUND OF THE INVENTION

This application claims the benefit of U.S. Provisional Application No. 60/135,221, filed May 21, 1999.

The present invention provides a software architecture that enables core software of a television set-top terminal to be compatible with different operating environments.

The recent advent of digital set-top terminals has spurred the growth of subscriber television networks, such as cable/satellite television networks. Such terminals can support increased levels of programming services and a variety of software-based applications and functions, such as an electronic program guide, stock or weather banners, shop and bank at home services, games, and the like. Moreover, this trend is expected to continue with the convergence of telephone, television and computer networks, and the rise of in-home computer networks.

A digital set-top box (e.g., subscriber terminal) for cable or satellite television, or the like, requires OS software, middleware and device drivers to function.

OS software provides the multithreading, real-time OS support needed to operate the set-top.

Middleware is software that connects two otherwise

separate applications. For example, such middleware can be provided to mediate between an application program and a network, thereby managing the interaction between disparate applications across heterogeneous computing platforms. More generally, middleware serves as the glue between separate applications, and is sometimes referred to as "plumbing" because it connects two sides of an application and passes data between them.

Moreover, drivers are used to control different hardware devices in a terminal, such as tuners, demodulators, MPEG-2 decoders (e.g., audio, video, and data), video encoders, audio mixers, and so forth.

In a set-top box, "core software" is provided that allows the set-top to provide such necessary television functions as service acquisition, system information (SI) management, download capability (e.g., for new application and OS software), return path communication (e.g., for polling the set-top for billing purposes), set-top configuration, and conditional access control (i.e., security).

In the past, each set-top has only been designed to use one particular operating environment. The problem of handling multiple operating environments has not previously been addressed in the set-top environment. Generally, applications have been ported to various operating environments either through rewrite of the set-top system software to that environment or through an abstraction layer.

Accordingly, it would be advantageous to enable a choice of different operating environments on a set-top without the need to change the core set-top system software. This would enable the reuse of a single set-

top design for multiple operating environments. It would also enable faster implementation of alternate operating software environments, permitting a faster time to market new set-top devices.

In particular, it would be desirable to have a software architecture that allows a customer, such as a cable television system operator, to choose the operating environment, including the OS kernel, middleware and application software, and device drivers, that will be used in the set-top. The core set-top system software should allow the device to operate within a common system environment. The architecture should allow for the development of the core software, and then implementation of the architecture in any of a plurality of different operating environments, such as VRTX, Windows CE, AperiOS, PowerTV and other set-top operating environments.

The architecture should be suitable for terminals that receive programming services via any type of network, including broadband communications networks, Digital Subscriber Loop (DSL) networks, and others.

The present invention provides a software architecture that enjoys the aforementioned and other advantages.

SUMMARY OF THE INVENTION

In accordance with the present invention, a software architecture is provided to enable core software of a set-top device to operate in any of a plurality of different operating environments.

In a particular embodiment, an apparatus for providing a software architecture for implementing a television subscriber terminal includes a computer readable medium having computer program code means, and means for executing the computer program code means to implement a layered software architecture. The architecture includes: a first, hardware layer, a second, device driver layer, a third, device driver interface layer, a fourth layer with the core system software for providing television functions, and a kernel abstraction function, a fifth layer for middleware porting, and a sixth layer providing middleware to interface with at least one application program (at a seventh layer).

The kernel abstraction layer and the device driver interface layer enable the core system software to operate in different operating environments. Moreover, an operating system (OS) kernel is provided at the fourth and fifth layers.

The architecture further includes a software interface between the core system software and the fifth layer that enables the core system software to operate in the different operating environments.

The television functions of the core system software include one or more of: message reception and distribution, system information processing, terminal

configuration, terminal control message processing, service acquisition, conditional access control, download capability, return path communication, and diagnostic data management.

Moreover, the television functions may include conditional access services, including one or more of: object authentication, object authorization, resource authorization, baseline privacy key exchange services, and cryptographic functions.

A corresponding method is also presented.

BRIEF DESCRIPTION OF THE DRAWING

The Figure depicts a television subscriber terminal architecture in accordance with the invention.

DETAILED DESCRIPTION OF THE INVENTION

The Figure illustrates a software platform, or achitecture, for implementing a set-top box in accordance with the present invention. The platform can be implemented using any suitable operating system, such as the VRTX (virtual real-time executive) operating system available from Mentor Graphics Corporation of Wilsonville, Oregon, Windows CE available from Microsoft Corporation of Redmond, Washington, AperiOS available from Sony Corporation of New York, N.Y., PowerTV available from PowerTV, Inc. of Cupertino, CA, etc.

The architecture can be considered to include three main levels, including a lower level (functions 30, 32 and 34) related to hardware and device driver functions, a middle level (functions 14, 16, 18, 20, 22, 24 and 26) related to the core set-top software, and a high level (functions 10, 12) related to middleware and applications.

Specifically, the top level of the software structure comprises one or more application programs 10 which can comprise, for example, the Digital Television (DTV) navigator application (available from Liberate Technologies of San Carlos, California), interactive games, enhanced television features, Internet web browsers, etc. These applications need to communicate with the core set-top system software 20. This communication is accomplished with the middleware/application interface 12, the middleware porting layer 14, and the software interface 26.

The next layer in the software structure comprises

middleware/application interface APIs (application program interfaces) 12. This layer 12 is part of the operating environment (i.e., it is OS-specific), and provides the interface to the application program(s) 10 which execute in the set-top (e.g., cable set-top hardware 34). The layer 12 provides facilities for applications to use and control the set-top box resources, and to coexist with other applications. The layer 12 also provides the environment in which the applications execute. Layer 12 also provides the API to which applications are written, providing an authoring environment. Moreover, this layer 12 handles the issues of shared resources, multiple applications, application selection, application launch and application termination. Preferably, it provides a consistent interface to the end user.

A middleware porting layer 14, e.g., for cable, DSL or other television services, is provided to translate the television, graphics and communication interfaces 20, 22 and 24, respectively, to the particular functional interface required by a middleware software component and specified by the middleware provider/vendor. Middleware providers often specify an interface on which their software has been tested and proven. This layer 14, which is also part of the operating environment, allows portability of the middleware solution to the set-top in an expedient manner. More specifically, the layer 14 allows the core set-top system software 20 to remain the same while allowing different middleware layers to execute.

An operating system (OS) kernel 16 is provided at the levels of the middleware porting layer 14 and the

core set-top system software 20, the graphics, audio, video subsystem 22 and the network communication stacks 24. This kernel 16 can comprise, for example, a MIPS (million instructions per second) port of the VRTX kernel version 4.0, available from the Microtec division of Mentor Graphics Corporation for the Motorola 68000 family of microprocessors. This kernel 16 provides the multithreading, real time OS support needed to operate the set-top. The OS kernel 16, together with the middleware porting layer 14 and middleware/application interface APIs 12, and the device drivers are components of the operating environment. More particularly, the OS kernel 16 provides OS facilities such as multitasking, inter-task communication, synchronization, and memory management. These facilities are used to create tasks, set task priorities, communicate between tasks, and synchronize operation of the system tasks and application tasks.

A kernel abstraction component 18 in accordance with the present invention provides generic kernel functions needed by the other components. In particular, component 18 abstracts these functions from the specifics of the OS kernel 16. The kernel abstraction component 18 includes two sections, namely, (i) the kernel calls used by the components, and (ii) the kernel-specific translations. Some of the supported kernel features are implemented in a fashion that is independent of the underlying kernel.

In accordance with the present invention, the kernel abstraction component 18 separates the details of the OS kernel from the core set-top system software 20, and is used to isolate the core system software from the

specifics of an OS kernel implementation. In this manner, portability of the core software 20 between operating environments is provided. The OS kernel abstraction 18 provides standard types of OS services in a generic way to the core system software 20. These services include, e.g., messaging, thread operations, memory management, synchronization, and event management.

The core set-top system software 20 contains support for the television system features. For example, this software can be written to support the DigiCipher® II digital television system features. DigiCipher® II is a proprietary system developed by General Instrument Corporation of Horsham, Pennsylvania, U.S.A., the assignee hereof. This system includes features that are specific to the cable television environment for downstream and upstream communications. The features supported by the core set-top system software 20 include, without limitation, service acquisition (e.g., acquisition of a pay-per-view movie), system information management, set-top configuration and control, return path for polling, download capability including authentication and authorization, and non-volatile memory management.

The core set-top system software 20 also includes conditional access system drivers (CABIOS), which are used to provide security and access control. This component supports the features of object authentication, object authorization, resource authorization, cryptographic tool kit, and baseline privacy key exchange. The interface layer for the cryptographic tool kit (Crypto Toolkit) is the Public

Key Cryptography Standards (PKCS) 11 standard.

More specifically, the core set-top system software 20 provides the services that allow the set-top to exist in a communication (e.g., cable television) network. As noted above, the functions of this software include:

Message reception and distribution - This software component receives messages over a control channel or through an in-band channel, and distributes them to the appropriate software process for interpretation and processing.

System information processing - This component interprets the System Information provided by the protocol (e.g., Advanced Television Systems Committee (ATSC), or Digital Video Broadcast (DVB) standard) specifying the Modulation Type, Channel Map, Service Maps, Service Descriptions, Service Components. This information allows the software to find, reconstruct and deliver the service to the user or application.

Configuration and control message processing - This component interprets the configuration and control messages sent to the set-top from, e.g., a cable headend controller. This information is used to provision a set-top to operate within the cable television system or other applicable network. The component sets up the set-top's features and provides the ability to control the set-top operation remotely by the system operator.

Digital and analog service acquisition - This component performs all of the necessary functions to 'tune' a video/audio service to be presented to the television viewer.

Conditional access subsystem - this component

authorizes services and applications. Video services are delivered to the set-top in an encrypted form. The Conditional Access (CA) system compares authorization information about the service with authorization information that is delivered to the set-top. When a service is deemed authorized, the CA system enables the decryption of that service.

Download subsystem - this component acquires software objects from the cable or other applicable network and stores them in the set-top's memory. These objects may include, e.g., software suites or applications. The download subsystem determines if it is authorized to accept the object and, if it is, then processes and stores each segment of the object. Once the object is stored, it is validated and can then be enabled to execute either from, e.g., the headend control, or by the operating system.

Return path subsystem - This component collects purchase and diagnostic information and transmits it back to the headend (or other appropriate system operator site) upon command. This information is ultimately sent to a billing system to enable billing of services used by the set-top. The return path can be a store-and-forward mechanism or an on-demand interactive mechanism.

Diagnostic Data management subsystem - This component collects and manages data which can be used to diagnose the state of the set-top, allowing for remote determination of the operation and health of the set-top software. This data is made available to the OS for display to the user or installer. It is also made

available to the return path subsystem for collection by a computer at the system operator's site (e.g., at a cable headend).

The CABIOS (Conditional Access Basic Services) provides conditional access services above the normal service access control, including:

Object Authentication - This function authenticates a software object to determine that the object is the intended object. Such authentication is accomplished, for example, by matching a signature delivered with the object.

Object Authorization - This function authorizes objects to be loaded and executed in the set-top. It uses the conditional access hardware to determine the authorization status of the set-top.

Resource Authorization - This function authorizes the use of set-top resources through the conditional access hardware.

Baseline Privacy Key Exchange Services - These functions support the key exchange requirements of the Cable modem system, which may conform to the Data-Over-Cable Service Interface Specifications (DOCSIS).

Cryptographic Functions - These functions provide a basic cryptographic toolkit for use by the application operating system and applications within the set-top. As indicated above, a typical cryptographic tool set follows the PKCS 11 standard, although other standards can be used.

The graphics, audio and video subsystem 22 includes, in a preferred embodiment, two-dimensional (2D), three-dimensional (3D), and video libraries

working on top of a graphics display driver. Such display drivers are commonly available from various vendors such as ATI Technologies Inc. of Thornhill, Ontario, Canada. Subsystem 22 provides a standard API to which middleware providers can be ported, and enables video to be viewed combined with graphics. More particularly, the analog or digital video is combined with a graphics overlay to present a combined view to the user. This subsystem 22 also provides audio, which can exist in multiple modes, such as stereo and surround sound. The audio subsystem can also support record and playback of audio files.

The network communications stacks 24 provide standardized networking stacks to facilitate web browsing and communications with external devices. This subsystem 24 includes many standardized network protocols which can be delivered over multiple communications paths. The stacks can comprise, for example, Mentor/Microtec's CNX communications extension to the VRTX operating system, available from Mentor Graphics Corporation of Wilsonville, Oregon.

This subsystem 24 resides on top of the communications drivers (e.g., Ethernet and DOCSIS drivers) and provides, for example, a Berkeley sockets (Berkeley Software Design, Inc.) interface to protocols such as Transmission Control Protocol (TCP)/Internet Protocol (IP) (e.g., a TCP/IP interface over a DOCSIS modem or over an Ethernet device), User Datagram Protocol (UDP), Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), Dynamic Host Configuration Protocol (DHCP) Client, Domain Name System (DNS) Client, and Point-to-Point Protocol (PPP) (dial-up

uses CCP, Link Control Protocol (LCP) and Internet Protocol Control Protocol (IPCP) protocols).

For the set-top hardware 34 to operate the set-top box for its intended functions, various drivers must be provided, as well known in the art. These are depicted in the Figure as low level device drivers 32, which, in accordance with the present invention, are interfaced to a driver model used by the core set-top system software 20 via device drivers-to-driver model interface 30.

The interface 30 separates the details of the driver model environment from the core system software 20. The interface 30 is used to isolate the device drivers 32 from the specifics of the driver model provided by the operating environment. The low level device driver software 32 is hardware-specific and portable across operating environments. The driver model interface 30 ties the hardware-specific device driver 32 to the OS by providing the OS's desired interface to the driver.

Device drivers are provided for each device supported on the set-top. These include a graphics driver, which may be written, e.g., on an ATI Hardware Access Provider (HAP) layer available from ATI Technologies, Inc. A graphics library for use in the set-top can written on the HAP. Other typical set-top device drivers include a communications driver, the tuner (e.g., a combination of the General Instrument DOCSIS tuner and Quadrature Amplitude Modulation (QAM) code), serial ports (using, e.g., an IEEE 1394 high performance serial bus), parallel ports, Ethernet ports, Universal Serial Bus (USB) ports, Light-Emitting Diodes (LEDs), Keypad and/or Keyboard drivers, a DOCSIS driver,

such as the Broadcom DOCSIS driver available from Broadcom Corporation of Irvine, California, U.S.A., and a smart card driver. A smart card may be used in the set-top for consumer purchasing applications such as on-line buying. Each device in the set-top requires a driver function to operate.

It should now be appreciated that the present invention solves the problem of enabling core set-top software to be compatible with multiple set-top operating environments. With the software architecture provided by the invention, it is not necessary to port the core software to different operating environments by rewriting code specific to each environment or through an abstraction layer. Instead, a unique device drivers-to-driver model interface is provided together with a kernel abstraction component that overcomes portability issues and provides for a clean interface between layers in the architecture. The invention can be implemented using object oriented techniques, such as C++ or Java programming, although the invention is by no means limited to such an implementation.

The combination of the components described herein provides a basis for set-top software to enable advanced features of an advanced set-top box implementation. Each of the components provides necessary functions required by an advanced set-top terminal. The novel architecture of the invention provides the benefits of reuse and portability of the core set-top software to alternate operating environments, thereby allowing well-understood and previously tested software components to be reused. The architecture also allows implementations in alternate operating environments to be quickly and

straightforwardly provided.

Although the invention has been described in connection with various specific implementations, it should be appreciated that various adaptations and modifications can be made thereto without departing from the scope of the invention as set forth in the claims.

What is claimed is:

1. An apparatus for providing a software architecture for implementing a television subscriber terminal, comprising:

a computer readable medium having computer program code means; and

means for executing said computer program code means to implement a layered software architecture including:

a first layer supporting hardware of the terminal;

a second layer comprising at least one device driver;

a third layer comprising an interface for the at least one device driver;

a fourth layer comprising core system software for providing television functions;

said fourth layer also including a kernel abstraction function which, together with said third layer, enables said core system software to operate in different operating environments;

a fifth layer for middleware porting;

an operating system kernel at said fourth and fifth layers; and

a sixth layer providing middleware to interface with at least one application program.

2. The apparatus of claim 1, wherein:

an operating environment comprises said middleware, middleware porting, and operating system kernel.

3. The apparatus of claim 1, wherein:

an operating environment comprises said middleware, middleware porting, operating system kernel, and device driver.

4. The apparatus of claim 1, wherein:
an operating environment comprises said middleware, middleware porting, operating system kernel, device driver, and application program.

5. The apparatus of claim 1, wherein:
the television subscriber terminal receives television signals via a broadband communication network.

6. The apparatus of claim 1, wherein the television functions include at least one of:
message reception and distribution;
system information processing;
terminal configuration;
terminal control message processing;
service acquisition;
conditional access control;
download capability;
return path communication; and
diagnostic data management.

7. The apparatus of claim 1, wherein the television functions include conditional access services, including at least one of:
object authentication;
object authorization;
resource authorization;

baseline privacy key exchange services; and cryptographic functions.

8. The apparatus of claim 1, wherein:

the kernel abstraction function provides kernel calls and kernel-specific translations for the operating system kernel.

9. The apparatus of claim 1, wherein:

the kernel abstraction function isolates the core system software from portions of the operating system kernel that are specific to an operating environment.

10. The apparatus of claim 1, wherein:

the interface for the at least one device driver isolates the core system software from portions of the device driver that are specific to an operating environment.

11. The apparatus of claim 1, wherein:

the operating system kernel provides multi-threading, real-time operating system support needed to operate the terminal.

12. The apparatus of claim 1, wherein:

the operating system kernel provides facilities for the terminal that are specific to an operating environment, including at least one of: multi-tasking, inter-task communication, messaging, synchronization, memory management, and event management.

13. The apparatus of claim 1, wherein:

the architecture further includes a software interface between the core system software and the fifth layer that enables said core system software to operate in the different operating environments.

14. The apparatus of claim 13, wherein:

the software interface isolates the core system software from portions of the middleware and middleware porting that are specific to an operating environment.

15. The apparatus of claim 13, wherein:

the software interface isolates the core system software from portions of the application program that are specific to an operating environment.

16. A method for providing a software architecture for implementing a television subscriber terminal, comprising the steps of:

providing a computer readable medium having computer program code means; and

executing said computer program code means to implement a layered software architecture including:

a first layer supporting hardware of the terminal;

a second layer comprising at least one device driver;

a third layer comprising an interface for the at least one device driver;

a fourth layer comprising core system software for providing television functions;

said fourth layer also including a kernel abstraction function which, together with said third layer, enables said core system software to operate in

different operating environments;

a fifth layer for middleware porting;

an operating system kernel at said fourth and fifth layers; and

a sixth layer providing middleware to interface with at least one application program.

17. The method of claim 16, wherein:

an operating environment comprises said middleware, middleware porting, and operating system kernel.

18. The method of claim 16, wherein:

an operating environment comprises said middleware, middleware porting, operating system kernel, and device driver.

19. The method of claim 16, wherein:

an operating environment comprises said middleware, middleware porting, operating system kernel, device driver, and application program.

20. The method of claim 16, wherein:

the television subscriber terminal receives television signals via a broadband communication network.

21. The method of claim 16, wherein the television functions include at least one of:

message reception and distribution;

system information processing;

terminal configuration;

terminal control message processing;

service acquisition;
conditional access control;
download capability;
return path communication; and
diagnostic data management.

22. The method of claim 16, wherein the television functions include conditional access services, including at least one of:
object authentication;
object authorization;
resource authorization;
baseline privacy key exchange services; and
cryptographic functions.

23. The method of claim 16, wherein:
the kernel abstraction function provides kernel calls and kernel-specific translations for the operating system kernel.

24. The method of claim 16, wherein:
the kernel abstraction function isolates the core system software from portions of the operating system kernel that are specific to an operating environment.

25. The method of claim 16, wherein:
the interface for the at least one device driver isolates the core system software from portions of the device driver that are specific to an operating environment.

26. The method of claim 16, wherein:

the operating system kernel provides multi-threading, real-time operating system support needed to operate the terminal.

27. The method of claim 16, wherein:

the operating system kernel provides facilities for the terminal that are specific to an operating environment, including at least one of: multi-tasking, inter-task communication, messaging, synchronization, memory management, and event management.

28. The method of claim 16, wherein:

the architecture further includes a software interface between the core system software and the fifth layer that enables said core system software to operate in the different operating environments.

29. The method of claim 28, wherein:

the software interface isolates the core system software from portions of the middleware and middleware porting that are specific to an operating environment.

30. The method of claim 28, wherein:

the software interface isolates the core system software from portions of the application program that are specific to an operating environment.

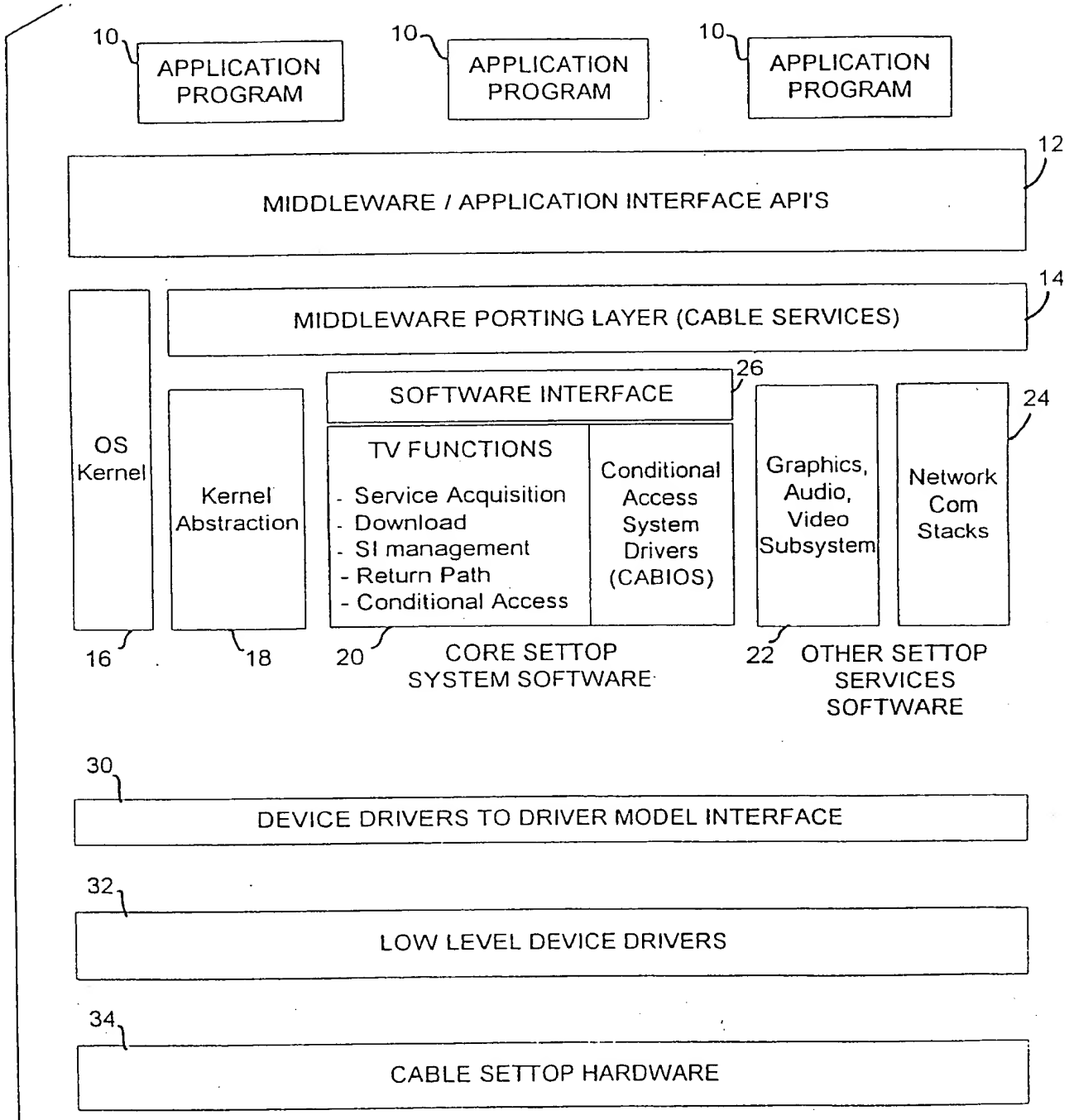

10
APPLICATION PROGRAM
10
APPLICATION PROGRAM
10
APPLICATION PROGRAM
12
MIDDLEWARE / APPLICATION INTERFACE API'S
14
MIDDLEWARE PORTING LAYER (CABLE SERVICES)
OS Kernel
Kernel Abstraction
26
SOFTWARE INTERFACE
TV FUNCTIONS
- Service Acquisition
- Download
- SI management
- Return Path
- Conditional Access
Conditional Access System Drivers (CABIOS)
Graphics, Audio, Video Subsystem
24
Network Com Stacks
16
18
20
CORE SETTOP SYSTEM SOFTWARE
22
OTHER SETTOP SERVICES SOFTWARE
30
DEVICE DRIVERS TO DRIVER MODEL INTERFACE
32
LOW LEVEL DEVICE DRIVERS
34
CABLE SETTOP HARDWARE

FIG. 1

INTERNATIONAL SEARCH REPORT

Interr ıal Application No
PCT/US 00/07980

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N5/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 909 094 A (CANAL PLUS SA) 14 April 1999 (1999-04-14) column 6, line 1 -column 8, line 48 abstract; figure 3	1-30
X	HURLEY T R: "EVOLUTION OF THE DIGITAL SET TOP BOX" INTERNATIONAL BROADCASTING CONVENTION,GB,LONDON, no. 428, 1996, pages 277-282, XP000905436 section "Software architecture and interactive services" figure 4 -/--	1-30

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 July 2000	09/08/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	La, V

1

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
P,X	SUN MICROSYSTEMS, INC: "Java TV API Specification" DRAFT 0.9.0, 'Online! 14 June 1999 (1999-06-14), XP002143725 Retrieved from the Internet: <URL:http://java.sun.com/products/javatv/javatvspec0_9_0.html> 'retrieved on 2000-07-27! section 1. "Introduction" Introduction of subsection 2.2 "Software environment" and figure 2 subsection 2.2.2.3 "Security"	1-30
A	EP 0 813 147 A (LSI LOGIC CORP) 17 December 1997 (1997-12-17) column 3, line 11 -column 5, line 9 column 5, line 31 -column 6, line 46 figures 1-3	1-30
A	ITOH J -I ET AL: "A NEW SOFTWARE ARCHITECTURE FOR EVOLVABLE MULTIMEDIA SOFTWARE" EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS,IT,EUREL PUBLICATION, MILANO, vol. 8, no. 4, 1 July 1997 (1997-07-01), pages 423-435, XP000695449 ISSN: 1124-318X the whole document	1-30
A	THRIFT P ET AL: "JTV - Java-enabled Television" PROCEEDINGS OF THE SPIE, 4 November 1997 (1997-11-04), XP002090497 section 1. "Introduction" section 2. "JTV API Overview" abstract; figure 1	1-30
A	EVAIN J -P: "THE MULTIMEDIA HOME PLATFORM" EBU REVIEW- TECHNICAL,BE,EUROPEAN BROADCASTING UNION. BRUSSELS, no. 275, 21 March 1998 (1998-03-21), pages 4-10, XP000767493 ISSN: 0251-0936 the whole document	1-30

-/--

1

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	WILLIAMS T: "STB OPERATING SYSTEMS GEAR UP FOR FLOOD OF DATA SERVICES" COMPUTER DESIGN,US,PENNWELL PUBL. LITTLETON, MASSACHUSETTS, vol. 35, no. 2, 1 February 1996 (1996-02-01), pages 67-68,72,74-76, XP000555483 ISSN: 0010-4566 the whole document	1-30
A	B. MARSTELLER AND R. BAER: "Sun announces key consumer technology to bring the power of the JAVA platform to auto, TV and phone markets" PRESS RELEASE, 'Online! 24 March 1998 (1998-03-24), XP002143726 Retrieved from the Internet: <URL:http://www.javasoft.com/pr/1998/03/pr980324-04.html> 'retrieved on 2000-07-27! section "JavaTV API"	1-30

1

INTERNATIONAL SEARCH REPORT

ormation on patent family members

Interr ·al Application No
PCT/US 00/07980

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0909094 A	14-04-1999	AU 9363298 A	27-04-1999
		EP 1021918 A	26-07-2000
		WO 9918730 A	15-04-1999
		ZA 9809081 A	13-04-1999
EP 0813147 A	17-12-1997	US 5946487 A	31-08-1999
		JP 10069394 A	10-03-1998

INTERNATIONAL SEARCH REPORT

Information on patent family members

Interr[illegible]nal Application No
PCT/US 00/07980

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0909094 A	14-04-1999	AU 9363298 A	27-04-1999
		EP 1021918 A	26-07-2000
		WO 9918730 A	15-04-1999
		ZA 9809081 A	13-04-1999
EP 0813147 A	17-12-1997	US 5946487 A	31-08-1999
		JP 10069394 A	10-03-1998

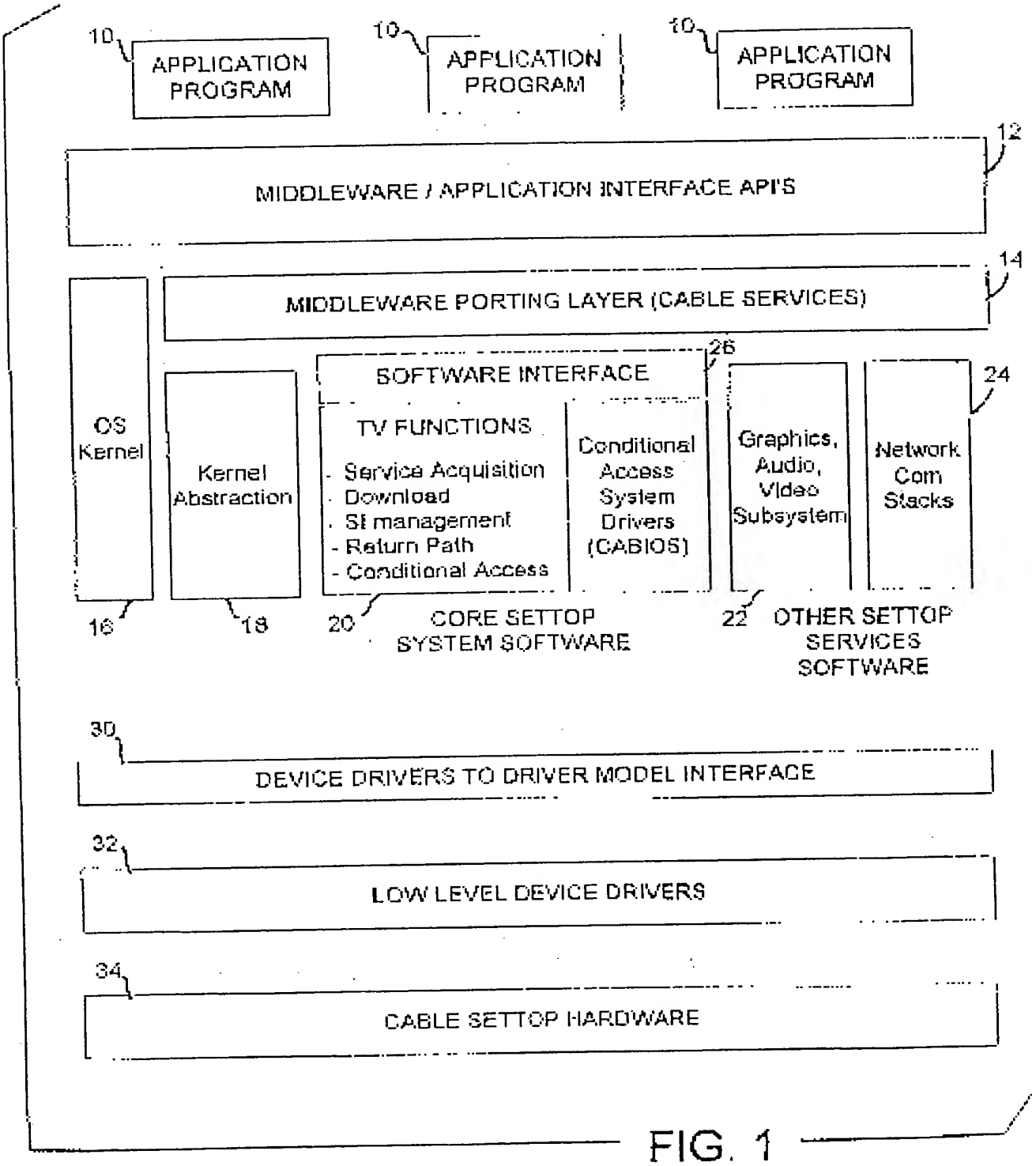


FIG. 1